Multi-objective Model Checking of Markov Decision Processes

Etessami, K., Kwiatkowska, M. Z., Vardi, M. Y., & Yannakakis, M. (2007). Multi-objective Model Checking of Markov Decision Processes. In TACAS. (pp. 50-65). Springer Berlin / Heidelberg.

Multi-Objective Model Checking of Markov Decision Processes

K. Etessami[1], M. Kwiatkowska[2], M. Y. Vardi[3], and M. Yannakakis[4]

[1] LFCS, School of Informatics, University of Edinburgh
[2] School of Computer Science, Birmingham University
[3] Dept. of Computer Science, Rice University
[4] Dept. of Computer Science, Columbia University

Abstract. We study and provide efficient algorithms for multi-objective model checking problems for Markov Decision Processes (MDPs). Given an MDP, M, and given multiple linear-time (ω-regular or LTL) properties φ_i, and probabilities $r_i \in [0,1]$, $i = 1, \ldots, k$, we ask whether there exists a strategy σ for the controller such that, for all i, the probability that a trajectory of M controlled by σ satisfies φ_i is at least r_i. We provide an algorithm that decides whether there exists such a strategy and if so produces it, and which runs in time polynomial in the size of the MDP. Such a strategy may require the use of both randomization and memory. We also consider more general multi-objective ω-regular queries, which we motivate with an application to assume-guarantee compositional reasoning for probabilistic systems.

Note that there can be trade-offs between different properties: satisfying property φ_1 with high probability may necessitate satisfying φ_2 with low probability. Viewing this as a multi-objective optimization problem, we want information about the "trade-off curve" or *Pareto curve* for maximizing the probabilities of different properties. We show that one can compute an approximate Pareto curve with respect to a set of ω-regular properties in time polynomial in the size of the MDP.

Our quantitative upper bounds use LP methods. We also study qualitative multi-objective model checking problems, and we show that these can be analysed by purely graph-theoretic methods, even though the strategies may still require both randomization and memory.

1 Introduction

Markov Decision Processes (MDPs) are standard models for stochastic optimization and for modelling systems with probabilistic and nondeterministic or controlled behavior (see [Put94,Var85,CY95,CY98]). In an MDP, at each state, the controller can choose from among a number of actions, or choose a probability distribution over actions. Each action at a state determines a probability distribution on the next state. Fixing an initial state and fixing the controller's strategy determines a probability space of infinite runs (trajectories) of the MDP. For MDPs with a single objective, the controller's goal is to optimize the value of an objective function, or payoff, which is a function of the entire trajectory.

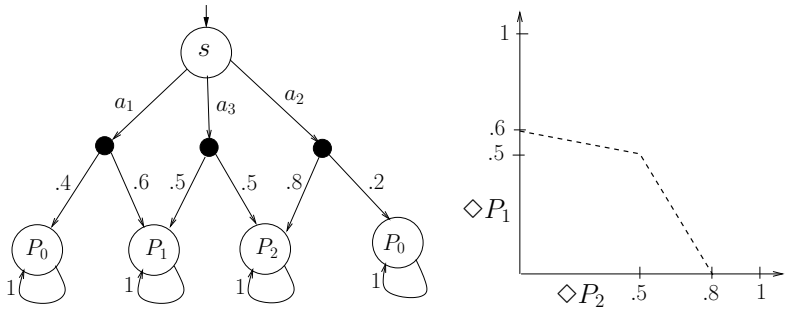

Fig. 1. An MDP with two objectives, $\Diamond P_1$ and $\Diamond P_2$, and the associated Pareto curve.

Many different objectives have been studied for MDPs, with a wide variety of applications. In particular, in verification research linear-time model checking of MDPs has been studied, where the objective is to maximize the probability that the trajectory satisfies a given ω-regular or LTL property ([CY98,CY95,Var85]).

In many settings we may not just care about a single property. Rather, we may have a number of different properties and we may want to know whether we can simultaneously satisfy all of them with given probabilities. For example, in a system with a server and two clients, we may want to maximize the probability for both clients 1 and 2 of the temporal property: "every request issued by client i eventually receives a response from the server", $i = 1, 2$. Clearly, there may be a trade-off. To increase this probability for client 1 we may have to decrease it for client 2, and vice versa. We thus want to know what are the simultaneously *achievable* pairs (p_1, p_2) of probabilities for the two properties. More specifically, we will be interested in the "trade-off curve" or *Pareto curve*. The Pareto curve is the set of all achievable vectors $p = (p_1, p_2) \in [0, 1]^2$ such that there does not exist another achievable vector p' that *dominates* p, meaning that $p \leq p'$ (coordinate-wise inequality) and $p \neq p'$.

Concretely, consider the very simple MDP depicted in Figure 1. Starting at state s, we can take one of three possible actions $\{a_1, a_2, a_3\}$. Suppose we are interested in LTL properties $\Diamond P_1$ and $\Diamond P_2$. Thus we want to maximize the probability of reaching the two distinct vertices labeled by P_1 and P_2, respectively. To maximize the probability of $\Diamond P_1$ we should take action a_1, thus reaching P_1 with probability 0.6 and P_2 with probability 0. To maximize the probability of $\Diamond P_2$ we should take a_2, reaching P_2 with probability 0.8 and P_1 with probability 0. To maximize the *sum* total probability of reaching P_1 or P_2, we should take a_3, reaching both with probability 0.5. Now observe that we can also "mix" these pure strategies using randomization to obtain any convex combination of these three value vectors. In the graph on the right in Figure 1, the dotted line plots the Pareto curve for these two properties.

The Pareto curve $\mathcal{P}$ in general contains infinitely many points, and it can be too costly to compute an exact representation for it (see Section 2). Instead of

computing it outright we can try to *approximate* it ([PY00]). An ϵ-*approximate Pareto curve* is a set of achievable vectors $\mathcal{P}(\epsilon)$ such that for every achievable vector r there is some vector $t \in \mathcal{P}(\epsilon)$ which "almost" dominates it, meaning $r \leq (1+\epsilon)t$.

In general, given a labeled MDP M, k distinct ω-regular properties, $\Phi = \langle \varphi_i \mid i = 1, \ldots, k \rangle$, a start state u, and a strategy σ, let $\Pr_u^\sigma(\varphi_i)$ denote the probability that starting at u, using strategy σ, the trajectory satisfies φ_i. For a strategy σ, define the vector $t^\sigma = (t_1^\sigma, \ldots, t_k^\sigma)$, where $t_i^\sigma = \Pr_u^\sigma(\varphi_i)$, for $i = 1, \ldots, k$. We say a value vector $r \in [0,1]^k$ is *achievable* for Φ, if there exists a strategy σ such that $t^\sigma \geq r$.

We provide an algorithm that given MDP M, start state u, properties Φ, and rational value vector $r \in [0,1]^k$, decides whether r is achievable, and if so produces a strategy σ such that $t^\sigma \geq r$. The algorithm runs in time polynomial in the size of the MDP. The strategies may require both randomization and memory. Our algorithm works by first reducing the achievability problem for multiple ω-regular properties to one with multiple reachability objectives, and then reducing the multi-objective reachability problem to a multi-objective linear programming problem. We also show that one can compute an ϵ-approximate Pareto curve for Φ in time polynomial in the size of the MDP and in $1/\epsilon$. To do this, we use our linear programming characterization for achievability, and use results from [PY00] on approximating the Pareto curve for multi-objective linear programming problems.

We also consider more general *multi-objective queries*. Given a boolean combination B of quantitative predicates of the form $\Pr_u^\sigma(\varphi_i) \Delta p$, where $\Delta \in \{\leq, \geq, <, >, =, \neq\}$, and $p \in [0,1]$, a *multi-objective query* asks whether there exists a strategy σ satisfying B (or whether *all* strategies σ satisfy B). It turns out that such queries are not really much more expressive than checking achievability. Namely, checking a fixed query B can be reduced to checking a fixed number of *extended achievability* queries, where for some of the coordinates t_i^σ we can ask for a strict inequality, i.e., that $t_i^\sigma > r_i$. (In general, however, the number and size of the extended achievability queries needed may be exponential in the size of B.) A motivation for allowing general multi-objective queries is to enable *assume-guarantee compositional reasoning* for probabilistic systems, as explained in Section 2.

Whereas our algorithms for quantitative problems use LP methods, we also consider qualitative multi-objective queries. These are queries given by boolean combinations of predicates of the form $\Pr_u^\sigma(\varphi_i) \Delta b$, where $b \in \{0,1\}$. We give an algorithm using purely graph-theoretic techniques that decides whether there is a strategy that satisfies a qualitative multi-objective query, and if so produces such a strategy. The algorithm runs in time polynomial in the size of the MDP. Even for satisfying qualitative queries the strategy may need to use both randomization and memory.

In typical applications, the MDP is far larger than the size of the query. Also, ω-regular properties can be presented in many ways, and it was already shown in [CY95] that the query complexity of model checking MDPs against even a

single LTL property is 2EXPTIME-complete. We remark here that, if properties are expressed via LTL formulas, then our algorithms run in polynomial time in the size of the MDP and in 2EXPTIME in the size of the query, for deciding arbitrary multi-objective queries, where both the MDP and the query are part of the input. So, the worst-case upper bound is the same as with a single LTL objective. However, to keep our complexity analysis simple, we focus in this paper on the model complexity of our algorithms, rather than their query complexity or combined complexity.

Due to lack of space in the proceedings, many proofs have been omitted. Please see [EKVY07] for a fuller version of this paper, containing an appendix with proofs.

Related work. Model checking of MDPs with a single ω-regular objective has been studied in detail (see [CY98,CY95,Var85]). In [CY98], Courcoubetis and Yannakakis also considered MDPs with a single objective given by a positive weighted sum of the probabilities of multiple ω-regular properties, and they showed how to efficiently optimize such objectives for MDPs. They did not consider tradeoffs between multiple ω-regular objectives. We employ and build on techniques developed in [CY98].

Multi-objective optimization is a subject of intensive study in Operations Research and related fields (see, e.g., [Ehr05,Clí97]). Approximating the Pareto curve for general multi-objective optimization problems was considered by Papadimitriou and Yannakakis in [PY00]. Among other results, [PY00] showed that for multi-objective linear programming (i.e., linear constraints and multiple linear objectives), one can compute a (polynomial sized) ϵ-approximate Pareto curve in time polynomial in the size of the LP and in $1/\epsilon$.

Our work is related to recent work by Chatterjee, Majumdar, and Henzinger ([CMH06]), who considered MDPs with multiple discounted reward objectives. They showed that randomized but memoryless strategies suffice for obtaining any achievable value vector for these objectives, and they reduced the multi-objective optimization and achievability (what they call *Pareto realizability*) problems for MDPs with discounted rewards to multi-objective linear programming. They were thus able to apply the results of [PY00] in order to approximate the Pareto curve for this problem. We work in an undiscounted setting, where objectives can be arbitrary ω-regular properties. In our setting, strategies may require both randomization and memory in order to achieve a given value vector. As described earlier, our algorithms first reduce multi-objective ω-regular problems to multi-objective reachability problems, and we then solve multi-objective reachability problems by reducing them to multi-objective LP. For multi-objective reachabilility, we show randomized memoryless strategies do suffice. Our LP methods for multi-objective reachability are closely related to the LP methods used in [CMH06] (and see also, e.g., [Put94], Theorem 6.9.1., where a related result about discounted MDPs is established). However, in order to establish the results in our undiscounted setting, even for reachability we have to overcome some new obstacles that do not arise in the discounted case. In particular, whereas the "discounted frequencies" used in [CMH06] are always well-defined finite val-

ues under all strategies, the analogous undiscounted frequencies or "expected number of visits" can in general be infinite for an arbitrary strategy. This forces us to preprocess the MDPs in such a way that ensures that a certain family of undiscounted stochastic flow equations has a finite solution which corresponds to the "expected number of visits" at each state-action pair under a given (memoryless) strategy. It also forces us to give a quite different proof that memoryless strategies suffice to achieve any achievable vector for multi-objective reachability, based on the convexity of the memorylessly achievable set.

Multi-objective MDPs have also been studied extensively in the OR and stochastic control literature (see e.g. [Fur80,Whi82,Hen83,Gho90,WT98]). Much of this work is typically concerned with discounted reward or long-run average reward models, and does not focus on the complexity of algorithms. None of this work seems to directly imply even our result that for multiple reachability objectives checking achievability of a value vector can be decided in polynomial time, not to mention the more general results for multi-objective model checking.

2 Basics and background

A finite-state MDP $M = (V, \Gamma, \delta)$ consists of a finite set V of states, an action alphabet Γ, and a transition relation δ. Associated with each state v is a set of enabled actions $\Gamma_v \subseteq \Gamma$. The transition relation is given by $\delta \subseteq V \times \Gamma \times [0,1] \times V$. For each state $v \in V$, each enabled action $a \in \Gamma_v$, and every state $v' \in V$, we have exactly one transition $(v, \gamma, p_{(v,\gamma,v')}, v') \in \delta$, for some probability $p_{(v,\gamma,v')} \in [0,1]$, such that $\sum_{v' \in V} p_{(v,\gamma,v')} = 1$. Thus, at each state, each enabled action determines a probability distribution on the next state. There are no other transitions, so no transtitions on disabled actions. We assume every state v has some enabled action, i.e., $\Gamma_v \neq \emptyset$, so there are no dead ends. For our complexity analysis, we assume of course that all probabilities $p_{(v,\gamma,v')}$ are rational. A labeled MDP $M = (V, \Gamma, \delta, l)$ has, in addition a set of propositional predicates $Q = \{Q_1, \ldots, Q_r\}$ which label the states. We view this as being given by a labelling function $l : V \mapsto \Sigma$, where $\Sigma = 2^Q$. There are other ways to present MDPs, e.g., by separating controlled and probabilistic nodes into distinct states. The different presentations are equivalent and efficiently translatable to each other. For a labeled MDP $M = (V, \Gamma, \delta, l)$ with a given initial state $u \in V$, which we denote by M_u, runs of M_u are infinite sequences of states $\pi = \pi_0 \pi_1 \ldots \in V^\omega$, where $\pi_0 = u$ and for all $i \geq 0$, $\pi_i \in V$ and there is a transition $(\pi_i, \gamma, p, \pi_{i+1}) \in \delta$, for some $\gamma \in \Gamma_{\pi_i}$ and some probability $p > 0$. Each run induces an ω-word over Σ, namely $l(\pi) \doteq l(\pi_0) l(\pi_1) \ldots \in \Sigma^\omega$.

A *strategy* is a function $\sigma : (V\Gamma)^* V \mapsto \mathcal{D}(\Gamma)$, which maps a finite history of play to a probability distribution on the next action. Here $\mathcal{D}(\Gamma)$ denotes the set of probability distributions on the set Γ. Moreover, it must be the case that for all histories wu, $\sigma(wu) \in \mathcal{D}(\Gamma_u)$, i.e., the probabilty distribution has support only over the actions available at state u. A strategy is *pure* if $\sigma(wu)$ has support on exactly one action, i.e., with probability 1 a single action is played at every history. A strategy is *memoryless* (stationary) if the strategy depends only on the last state, i.e., if $\sigma(wu) = \sigma(w'u)$ for all $w, w' \in (V\Gamma)^*$. If σ is memoryless,

we can simply define it as a function $\sigma : V \mapsto \mathcal{D}(\Gamma)$. An MDP M with initial state u, together with a strategy σ, naturally induces a Markov chain M_u^σ, whose states are the histories of play in M_u, and such that from state $s = wv$ if $\gamma \in \Gamma_v$, there is a transition to state $s' = wv\gamma v'$ with probability $\sigma(wv)(\gamma) \cdot p_{(v,\gamma,v')}$. A run θ in M_u^σ is thus given by a sequence $\theta = \theta_0\theta_1 \ldots$, where $\theta_0 = u$ and each $\theta_i \in (V\Gamma)^*V$, for all $i \geq 0$. We associate to each history $\theta_i = wv$ the label of its last state v. In other words, we overload the notation and define $l(wv) \doteq l(v)$. We likewise associate with each run θ the ω-word $l(\theta) \doteq l(\theta_0)l(\theta_1)\ldots$. Suppose we are given φ, an LTL formula or Büchi automaton, or any other formalism for expressing an ω-regular language over alphabet Σ. Let $L(\varphi) \subseteq \Sigma^\omega$ denote the language expressed by φ. We write $\Pr_u^\sigma(\varphi)$ to denote the probability that a trajectory θ of M_u^σ satistifies φ, i.e., that $l(\theta) \in L(\varphi)$. For generality, rather than just allowing an initial vertex u we allow an initial probability distribution $\alpha \in \mathcal{D}(V)$. Let $\Pr_\alpha^\sigma(\varphi)$ denote the probability that under strategy σ, starting with initial distribution α, we will satify ω-regular property φ. These probabilities are well defined because the set of such runs is Borel measurable (see, e.g., [Var85,CY95]).

As in the introduction, for a k-tuple of ω-regular properties $\Phi = \langle\varphi_1, \ldots, \varphi_k\rangle$, given a strategy σ, we let $t^\sigma = (t_1^\sigma, \ldots, t_k^\sigma)$, with $t_i^\sigma = \Pr_u^\sigma(\varphi_i)$, for $i = 1, \ldots, k$. For MDP M and starting state u, we define the *achievable* set of value vectors with respect to Φ to be $U_{M_u,\Phi} = \{r \in \mathbb{R}^k_{\geq 0} \mid \exists\sigma \text{ such that } t^\sigma \geq r\}$. For a set $U \subseteq \mathbb{R}^k$, we define a subset $\mathcal{P} \subseteq U$ of it, called the *Pareto curve* or the *Pareto set* of U, consisting of the set of *Pareto optimal* (or *Pareto efficient*) vectors inside U. A vector $v \in U$ is called *Pareto optimal* if $\neg\exists v'(v' \in U \wedge v \leq v' \wedge v \neq v')$. Thus $\mathcal{P} = \{v \in U \mid v \text{ is Pareto optimal}\}$. We use $\mathcal{P}_{M_u,\Phi} \subseteq U_{M_u,\Phi}$ to denote the Pareto curve of $U_{M_u,\Phi}$.

It is clear, e.g., from Figure 1, that the Pareto curve is in general an infinite set. In fact, it follows from our results that for general ω-regular objectives the Pareto set is a convex polyhedral set. In principle, we may want to compute some kind of exact representation of this set by, e.g., enumerating all the vertices (on the upper envelope) of the polytope that defines the Pareto curve, or enumerating the facets that define it. It is not possible to do this in polynomial-time in general. In fact, the following theorem holds (the proof is omitted here):

Theorem 1. *There is a family of MDPs, $\langle M(n) \mid n \in \mathbb{N}\rangle$, where $M(n)$ has n states and size $O(n)$, such that for $M(n)$ the Pareto curve for two reachability objectives, $\Diamond P_1$ and $\Diamond P_2$, contains $n^{\Omega(\log n)}$ vertices (and thus $n^{\Omega(\log n)}$ facets).*

So, the Pareto curve is in general a polyhedral surface of superpolynomial size, and thus cannot be constructed exactly in polynomial time. We show, however, that the Pareto set can be efficiently *approximated* to any desired accuracy $\epsilon > 0$. An *ϵ-approximate Pareto curve*, $\mathcal{P}_{M_u,\Phi}(\epsilon) \subseteq U_{M_u,\Phi}$, is any achievable set such that $\forall r \in U_{M_u,\Phi}\ \exists t \in \mathcal{P}_{M_u,\Phi}(\epsilon)$ such that $r \leq (1+\epsilon)t$. When the subscripts M_u and Φ are clear from the context, we will drop them and use U, $\mathcal{P}$, and $\mathcal{P}(\epsilon)$ to denote the achievable set, Pareto set, and ϵ-approximate Pareto set, respectively.

We also consider general *multi-objective queries*. A *quantitative predicate* over ω-regular property φ_i is a statement of the form $\Pr_u^\sigma(\varphi_i)\Delta p$, for some rational

probability $p \in [0,1]$, and where Δ is a comparison operator $\Delta \in \{\leq, \geq, <, >, =\}$. Suppose B is a boolean combination over such predicates. Then, given M and u, and B, we can ask whether there exists a strategy σ such that B holds, or whether B holds for all σ. Note that since B can be put in DNF form, and the quantification over strategies pushed into the disjuction, and since ω-regular languages are closed under complementation, any query of the form $\exists \sigma B$ (or of the form $\forall \sigma B$) can be transformed to a disjunction (a negated disjunction, respectively) of queries of the form:

$$\exists \sigma \bigwedge_i (\Pr_u^\sigma(\varphi_i) \geq r_i) \wedge \bigwedge_j (\Pr_u^\sigma(\psi_j) > r'_j) \tag{1}$$

We call queries of the form (1) *extended achievability queries*. Thus, if the multi-objective query is fixed, it suffices to perform a fixed number of extended achievability queries to decide any multi-objective query. Note, however, that the number of extended achievability queries we need could be exponential in the size of B. We do not focus on optimizing query complexity in this paper.

A motivation for allowing general multi-objective queries is to enable *assume-guarantee compositional reasoning* for probabilistic systems. Consider, e.g., a probabilistic system consisting of the concurrent composition of two components, M_1 and M_2, where output from M_1 provides input to M_2 and thus controls M_2. We denote this by $M_1 \rhd M_2$. M_2 itself may generate outputs for some external device, and M_1 may also be controlled by external inputs. (One can also consider symmetric composition, where outputs from both components provide inputs to both. Here, for simplicity, we restrict ourselves to asymmetric composition where M_1 controls M_2.) Let M be an MDP with separate input and output action alphabets Σ_1 and Σ_2, and let φ_1 and φ_2 denote ω-regular properties over these two alphabets, respectively. We write $\langle \varphi_1 \rangle_{\geq r_1} M \langle \varphi_2 \rangle_{\geq r_2}$, to denote the assertion that *"if the input controller of M satisfies φ_1 with probability $\geq r_1$, then the output generated by M satisfies φ_2 with probability $\geq r_2$"*. Using this, we can formulate a general compositional assume-guarantee proof rule:

$$\frac{\langle \varphi_1 \rangle_{\geq r_1} M_1 \langle \varphi_2 \rangle_{\geq r_2} \quad \langle \varphi_2 \rangle_{\geq r_2} M_2 \langle \varphi_3 \rangle_{\geq r_3}}{\langle \varphi_1 \rangle_{\geq r_1}\ M_1 \rhd M_2\ \langle \varphi_3 \rangle_{\geq r_3}}$$

Thus, to check $\langle \varphi_1 \rangle_{\geq r_1} M_1 \rhd M_2 \langle \varphi_3 \rangle_{\geq r_3}$ it suffices to check two properties of smaller systems: $\langle \varphi_1 \rangle_{\geq r_1} M_1 \langle \varphi_2 \rangle_{\geq r_2}$ and $\langle \varphi_2 \rangle_{\geq r_2} M_2 \langle \varphi_3 \rangle_{\geq r_3}$. Note that checking $\langle \varphi_1 \rangle_{\geq r_1} M \langle \varphi_2 \rangle_{\geq r_2}$ amounts to checking that there does not exist a strategy σ controlling M such that $\Pr_u^\sigma(\varphi_1) \geq r_1$ and $\Pr_u^\sigma(\varphi_2) < r_2$.

We also consider *qualitative multi-objective queries*. These are queries restricted so that B contains only *qualitative predicates* of the form $\Pr_u^\sigma(\varphi_i) \Delta b$, where $b \in \{0,1\}$. These can, e.g., be used to check qualitative assume-guarantee conditions of the form: $\langle \varphi_1 \rangle_{\geq 1} M \langle \varphi_2 \rangle_{\geq 1}$. It is not hard to see that again, via

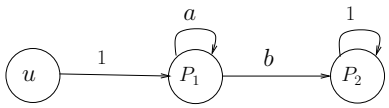

Fig. 2. The MDP M'.

boolean manipulations and complementation of automata, we can convert any qualitative query to a number of queries of the form:

$$\exists \sigma \bigwedge_{\varphi \in \Phi} (\Pr_u^\sigma(\varphi) = 1) \wedge \bigwedge_{\psi \in \Psi} (\Pr_u^\sigma(\psi) > 0)$$

where Φ and Ψ are sets of ω-regular properties. It thus suffices to consider only these qualitative queries.

In the next sections we study how to decide various classes of multi-objective queries, and how to approximate the Pareto curve for properties Φ. Let us observe here a difficulty that we will have to deal with. Namely, in general we will need both randomization and memory in our strategies in order to satisfy even simple qualitative multi-objective queries. Consider the MDP, M', shown in Figure 2, and consider the conjunctive query: $B \equiv \Pr_u^\sigma(\Box\Diamond P_1) > 0 \wedge \Pr_u^\sigma(\Box\Diamond P_2) > 0$. It is not hard to see that starting at state u in M' any strategy σ that satisfies B must use both memory and randomization. Each predicate in B can be satisfied in isolation (in fact with probability 1), but with a memoryless or deterministic strategy if we try to satisfy $\Box\Diamond P_2$ with non-zero probability, we will be forced to satisfy $\Box\Diamond P_1$ with probability 0. Note, however, that we can satisfy both with probability > 0 using a strategy that uses both memory and randomness: namely, upon reaching the state labeled P_1 for the first time, with probability $1/2$ we use move a and with probability $1/2$ we use move b. Thereafter, upon encountering the state labeled P_1 for the nth time, $n \geq 2$, we deterministically pick action a. This clearly assures that both predicates are satisfied with probability $= 1/2 > 0$.

3 Multi-objective reachability

In this section, as a step towards quantitative multi-objective model checking problems, we study a simpler multi-objective reachability problem. Specifically, we are given an MDP, $M = (V, \Gamma, \delta)$, a starting state u, and a collection of target sets $F_i \subseteq V$, $i = 1, \ldots, k$. The sets F_i may overlap. We have k objectives: the i-th objective is to maximize the probability of $\Diamond F_i$, i.e., of reaching some state in F_i. We assume that the states $F = \bigcup_{i=1}^k F_i$ are all absorbing states with a self-loop. In other words, for all $v \in F$, $(v, a, 1, v) \in \delta$ and $\Gamma_v = \{a\}$.[1]

[1] The assumption that target states are absorbing is necessary for the proofs in this section, but it will of course follow from the model checking results in Section 5, which build on this section, that multi-objective reachability problems for arbitrary target states can also be handled with the same complexities.

Objectives ($i = 1, \ldots, k$)**:** **Maximize** $\sum_{v \in F_i} y_v$;

Subject to:

$$
\begin{array}{lll}
\sum_{\gamma \in \Gamma_v} y_{(v,\gamma)} - \sum_{v' \in V} \sum_{\gamma' \in \Gamma_{v'}} p_{(v',\gamma',v)} y_{(v',\gamma')} = \alpha(v) & & \text{For all } v \in V \setminus F; \\
y_v - \sum_{v' \in V \setminus F} \sum_{\gamma' \in \Gamma_{v'}} p_{(v',\gamma',v)} y_{(v',\gamma')} & = 0 & \text{For all } v \in F; \\
y_v & \geq 0 & \text{For all } v \in F; \\
y_{(v,\gamma)} & \geq 0 & \text{For all } v \in V \setminus F \text{ and } \gamma \in \Gamma_u;
\end{array}
$$

Fig. 3. Multi-objective LP for the multi-objective MDP reachability problem

We first need to do some preprocessing on the MDP, to remove some useless states. For each state $v \in V \setminus F$ we can check easily whether there exists a strategy σ such that $Pr_v^\sigma(\Diamond F) > 0$: this just amounts to whether there exists a path from v to F in the underlying graph of the MDP. Let us call a state that does not satisfy this property a *bad* state. Clearly, for the purposes of optimizing reachability objectives, we can look for and remove all bad states from an MDP. Thus, it is safe to assume that bad states do not exist.[2] Let us call an MDP with goal states F *cleaned-up* if it does not contain any bad states.

Proposition 1. *For a cleaned-up MDP, an initial distribution $\alpha \in \mathcal{D}(V \setminus F)$, and a vector of probabilities $r \in [0,1]^k$, there exists a (memoryless) strategy σ such that $\bigwedge_{i=1}^k \Pr_\alpha^\sigma(\Diamond F_i) \geq r_i$ if and only if there exists a (respectively, memoryless) strategy σ' such that $\bigwedge_{i=1}^k \Pr_\alpha^{\sigma'}(\Diamond F_i) \geq r_i \ \wedge \ \bigwedge_{v \in V} \Pr_v^{\sigma'}(\Diamond F) > 0$.*

Now, consider the multi-objective LP described in Figure 3.[3] The set of variables in this LP are as follows: for each $v \in F$, there is a variable y_v, and for each $v \in V \setminus F$ and each $\gamma \in \Gamma_v$ there is a variable $y_{(v,\gamma)}$.

Theorem 2. *Suppose we are given a cleaned-up MDP, $M = (V, \Gamma, \delta)$ with multiple target sets $F_i \subseteq V$, $i = 1, \ldots, k$, where every target $v \in F = \bigcup_{i=1}^k F_i$ is an absorbing state. Let $\alpha \in \mathcal{D}(V \setminus F)$ be an initial distribution (in particular $V \setminus F \neq \emptyset$). Let $r \in (0,1]^k$ be a vector of positive probabilities. Then the following are all equivalent:*

[2] Technically, we would need to install a new "dead" absorbing state $v_{dead} \notin F$, such that all the probabilities going into states that have been removed now go to v_{dead}. For convenience in notation, instead of explicitly adding v_{dead} we treat it as implicit: we allow that for some states $v \in V$ and some action $a \in \Gamma_v$ we have $\sum_{v' \in V} p_{(v,\gamma,v')} < 1$, and we implicitly assume that there is an "invisible" transition to v_{dead} with the residual probability, i.e., with $p_{(v,\gamma,v_{dead})} = 1 - \sum_{v' \in V} p_{(v,\gamma,v')}$. Of course, v_{dead} would then be a "bad" state, but we can ignore this implicit state.

[3] We mention without further elaboration that this LP can be derived, using complementary slackness, from the dual LP of the standard LP for single-objective reachability obtained from Bellman's optimality equations, whose variables are x_v, for $v \in V$, and whose unique optimal solution is the vector x^* with $x_v^* = \max_\sigma \Pr_v^\sigma(\Diamond F)$ (see, e.g., [Put94,CY98]).

(1.) There is a (possibly randomized) memoryless strategy σ such that

$$\bigwedge_{i=1}^{k}(\Pr_\alpha^\sigma(\Diamond F_i) \geq r_i)$$

(2.) There is a feasible solution y' for the multi-objective LP in Fig. 3 such that

$$\bigwedge_{i=1}^{k}(\sum_{v\in F_i} y'_v \geq r_i)$$

(3.) There is an arbitrary strategy σ such that

$$\bigwedge_{i=1}^{k}(Pr_\alpha^\sigma(\Diamond F_i) \geq r_i)$$

Proof.
(1.) $\Rightarrow$ (2.). Since the MDP is cleaned up, by Proposition 1 we can assume there is a memoryless strategy σ such that $\bigwedge_{i=1}^{k} \Pr_\alpha^\sigma(\Diamond F_i) \geq r_i$ and $\forall v \in V$ $Pr_v^\sigma(\Diamond F) > 0$. Consider the square matrix P^σ whose size is $|V \setminus F| \times |V \setminus F|$, and whose rows and columns are indexed by states in $V \setminus F$. The (v, v')'th entry of P^σ, $P^\sigma_{v,v'}$, is the probability that starting in state v we shall in one step end up in state v'. In other words, $P^\sigma_{v,v'} = \sum_{\gamma\in\Gamma_v} \sigma(v)(\gamma) \cdot p_{v,\gamma,v'}$.

For all $v \in V \setminus F$, let $y'_{(v,\gamma)} = \sum_{v'\in V\setminus F} \alpha(v') \sum_{n=0}^{\infty}(P^\sigma)^n_{v',v}\sigma(v)(\gamma)$. In other words $y'_{(v,\gamma)}$ denotes the "expected number of times that, using the strategy σ, starting in the distribution α, we will visit the state v and upon doing so choose action γ". We don't know yet that these are finite values, but assuming they are, for $v \in F$, let $y'_v = \sum_{v'\in V\setminus F} \sum_{\gamma'\in\Gamma_{v'}} p_{(v',\gamma',v)} y'_{(v',\gamma')}$. This completes the definition of the entire vector y'.

Lemma 1. *The vector y' is well defined (i.e., all entries $y'_{(v,\gamma)}$ are finite). Moreover, y' is a feasible solution to the constraints of the LP in Figure 3.*

Now we argue that $\sum_{v\in F_i} y'_v = \Pr_\alpha^\sigma(\Diamond F_i)$. To see this, note that for $v \in F$, $y'_v = \sum_{v'\in V\setminus F} \sum_{\gamma'\in\Gamma_{v'}} p_{(v',\gamma',v)} y'_{(v',\gamma')}$ is precisely the "*expected number of times that we will transition into state v for the first time*", starting at distribution α. The reason we can say "for the first time" is because only the states in $V \setminus F$ are included in the matrix P^σ. But note that this italicised statement in quotes is another way to define the probability of eventually reaching state v. This equality can be establish formally, but we omit the formal algebraic derivation here. Thus $\sum_{v\in F_i} y'_v = \Pr_\alpha^\sigma(\Diamond F_i) \geq r_i$. We are done with (1.) $\Rightarrow$ (2.).

(2.) $\Rightarrow$ (1.). We now wish to show that if y'' is a feasible solution to the multi-objective LP such that $\sum_{v\in F_i} y''_v \geq r_i > 0$, for all $i = 1, \ldots, k$, then there exists a memoryless strategy σ such that $\bigwedge_{i=1}^{k} \Pr_\alpha^\sigma(\Diamond F_i) \geq r_i$.

Suppose we have such a solution y''. Let $S = \{v \in V \setminus F \mid \sum_{\gamma\in\Gamma_v} y''_{(v,\gamma)} > 0\}$. Let σ be the memoryless strategy, given as follows. For each $v \in S$

$$\sigma(v)(\gamma) := \frac{y''_{(v,\gamma)}}{\sum_{\gamma'\in\Gamma_v} y''_{v,\gamma'}}$$

Note that since $\sum_{\gamma \in \Gamma_v} y''_{(v,\gamma)} > 0$, $\sigma(v)$ is a well-defined probability distribution on the moves at state $v \in S$. For the remaining states $v \in (V \setminus F) \setminus S$, let $\sigma(v)$ be an arbitrary distribution in $\mathcal{D}(\Gamma_v)$.

Lemma 2. *This memoryless strategy σ satisfies $\bigwedge_{i=1}^{k} Pr_\alpha^\sigma(\Diamond F_i) \geq r_i$.*

Proof. The proof is in [EKVY07]. Here we very briefly sketch the argument. We can think of a feasible solution y'' to the LP constraints as defining a "stochastic flow", whose "source" is the initial distribution $\alpha(v)$, and whose sinks are F. By flow conservation, vertices $v \in V \setminus F$ that have positive outflow (and thus positive inflow) must all be reachable from the support of α, and must all reach F, and can not reach any vertex with zero outflow. The strategy σ is obtained by normalizing the outflow on each action at the states with positive outflow. It can be shown that, using σ, the expected number of times we choose action γ at vertex v is again given by $y''_{(v,\gamma)}$. Therefore, since transitions into the states $v \in F$ from $V \setminus F$ are only crossed once, the constraint defining the value y''_v yields $y''_v = \Pr_\alpha^\sigma(\Diamond\{v\})$. □

This completes the proof that (2.) ⇒ (1.).

(3.) ⇔ (1.). Clearly (1.) ⇒ (3.), so we need to show that (3.) ⇒ (1.).

Let U be the set of achievable vectors, i.e., all k-vectors $r = \langle r_1 \dots r_k \rangle$ such that there is a (unrestricted) strategy σ such that $\bigwedge_{i=1}^{k} \Pr_\alpha^\sigma(\Diamond F_i) \geq r_i$. Let $U^\odot$ be the analogous set where the strategy σ is restricted to be a possibly randomized but memoryless (stationary) strategy. Clearly, U and $U^\odot$ are both downward closed, i.e., if $r \geq r'$ and $r \in U$ then also $r' \in U$, and similarly with $U^\odot$. Also, obviously $U^\odot \subseteq U$. We characterized $U^\odot$ in (1.) ⇔ (2.), in terms of a multi-objective LP. Thus, $U^\odot$ is the projection of the feasible space of a set of linear inequalities (a polyhedral set), namely the set of inequalities in the variables y given in Fig. 3 and the inequalities $\sum_{v \in F_i} y_v \geq r_i$, $i = 1, \dots, k$. The feasible space is a polyhedron in the space indexed by the y variables and the r_i's, and $U^\odot$ is its projection on the subspace indexed by the r_i's. Since the projection of a convex set is convex, it follows that $U^\odot$ is convex.

Suppose that there is a point $r \in U \setminus U^\odot$. Since $U^\odot$ is convex, this implies that there is a separating hyperplane (see, e.g., [GLS93]) that separates r from $U^\odot$, and in fact since $U^\odot$ is downward closed, there is a separating hyperplane with non-negative coefficients, i.e. there is a non-negative "weight" vector $w = \langle w_1, \dots, w_k \rangle$ such that $w^T r = \sum_{i=1}^{k} w_i r_i > w^T x$ for every point $x \in U^\odot$.

Consider now the MDP M with the following undiscounted reward structure. There is 0 reward for every state, action and transition, except for transitions to a state $v \in F$ from a state in $V \setminus F$; i.e. a reward is produced only once, in the first transition into a state of F. The reward for every transition to a state $v \in F$ is $\sum \{w_i \mid i \in \{1, \dots, k\} \ \&\ v \in F_i\}$. By the definition, the expected reward of a policy σ is $\sum_{i=1}^{k} w_i \Pr_\alpha^\sigma(\Diamond F_i)$. From classical MDP theory, we know that there is a memoryless strategy (in fact even a deterministic one) that maximizes the expected reward for this type of reward structure. (Namely, this is a positive bounded reward case: see, e.g., Theorem 7.2.11 in [Put94].) Therefore,

$\max\{w^T x \mid x \in U\} = \max\{w^T x \mid x \in U^{\odot}\}$, contradicting our assumption that $w^T r > \max\{w^T x \mid x \in U^{\odot}\}$. □

Corollary 1. *Given an MDP $M = (V, \Gamma, \delta)$, a number of target sets $F_i \subseteq V$, $i = 1, \ldots, k + k'$, such that every state $v \in F = \bigcup_{i=1}^{k+k'} F_i$ is absorbing, and an initial state u (or even initial distribution $\alpha \in \mathcal{D}(V)$):*

(a.) Given an extended achievability query for reachability, $\exists\sigma B$, where

$$B \equiv \bigwedge_{i=1}^{k} (Pr_u^{\sigma}(\diamond F_i) \geq r_i) \wedge \bigwedge_{j=k+1}^{k+k'} (Pr_u^{\sigma}(\diamond F_j) > r_j),$$

we can in time polynomial in the size of the input, $|M|+|B|$, decide whether $\exists\sigma$ B is satisfiable and if so construct a memoryless strategy that satisfies it.

(b.) For $\epsilon > 0$, we can compute an ϵ-approximate Pareto curve $\mathcal{P}(\epsilon)$ for the multi-objective reachability problem with objectives $\diamond F_i$, $i = 1, \ldots, k$, in time polynomial in $|M|$ and $1/\epsilon$.

4 Qualitative multi-objective model checking

Theorem 3. *Given an MDP M, an initial state u, and a qualitative multi-objective query B, we can decide whether there exists a strategy σ that satisfies B, and if so construct such a strategy, in time polynomial in $|M|$, and using only graph-theoretic methods (in particular, without linear programming).*

Proof. (Sketch) By the discussion in Section 2, it suffices to consider the case where we are given MDP, M, and two sets of ω-regular properties Φ, Ψ, and we want a strategy σ such that

$$\bigwedge_{\varphi \in \Phi} \Pr_u^{\sigma}(\varphi) = 1 \wedge \bigwedge_{\psi \in \Psi} \Pr_u^{\sigma}(\psi) > 0$$

Assume the properties in Φ, Ψ are all given by (nondeterministic) Büchi automata A_i. We will use and build on results in [CY98]. In [CY98] (Lemma 4.4, page 1411) it is shown that we can construct from M and from a collection A_i, $i = 1, \ldots, m$, of Büchi automata, a new MDP M' (a refinement of M) which is the "product" of M with the *naive determinization* of all the A_i's (i.e., the result of applying the standard subset construction on each A_i, without imposing any acceptance condition).[4] This MDP M' has the following properties. For every subset R of $\Phi \cup \Psi$ there is a subset T_R of corresponding "target states" of M' (and we can compute this subset efficiently) that satisfies the following two conditions:

[4] Technically, we have to slightly adapt the constructions of [CY98], which use the convention that MDP states are either purely controlled or purely probabilistic, to the convention used in this paper which combines both control and probabilistic behavior at each state. But these adaptations are straightforward.

(I) If a trajectory of M' hits a state in T_R at some point, then we can apply from that point on a strategy μ_R (which is deterministic but uses memory) which ensures that the resulting infinite trajectory satisfies all properties in R almost surely (i.e., with conditional probability 1, conditioned on the initial prefix that hits T_R).

(II) For every strategy, the set of trajectories that satisfy all properties in R and do not infinitely often hit some state of T_R has probability 0.

We now outline the algorithm for deciding qualitative multi-objective queries.

1. Construct the MDP M' from M and from the properties Φ and Ψ.
2. Compute T_Φ, and compute for each property $\psi_i \in \Psi$ the set of states T_{R_i} where $R_i = \Phi \cup \{\psi_i\}$.[5]
3. If $\Phi \neq \emptyset$, prune M' by identifying and removing all "bad" states by applying the following rules.
   (a) All states v that cannot "reach" any state in T_Φ are "bad".[6]
   (b) If for a state v there is an action $\gamma \in \Gamma_v$ such that there is a transition $(v, \gamma, p, v') \in \delta$, $p > 0$, and v' is bad, then remove γ from Γ_v.
   (c) If for some state v, $\Gamma_v = \emptyset$, then mark v as bad.
   Keep applying these rules until no more states can be labelled bad and no more actions removed for any state.
4. Restrict M' to the reachable states (from the initial state u) that are not bad, and restrict their action sets to actions that have not been removed, and let M'' be the resulting MDP.
5. If ($M'' = \emptyset$ or $\exists \psi_i \in \Psi$ such that M'' does not contain any state of T_{R_i}) then return No.
   Else return Yes.

Correctness proof: In one direction, suppose there is a strategy σ such that $\bigwedge_{\varphi \in \Phi} \Pr_u^\sigma(\varphi) = 1 \wedge \bigwedge_{\psi \in \Psi} \Pr_u^\sigma(\psi) > 0$. First, note that there cannot be any finite prefix of a trajectory under σ that hits a state that cannot reach any state in T_Φ. For, if there was such a path, then all trajectories that start with this prefix go only finitely often through T_Φ. Hence (by property (II) above) almost all these trajectories do not satisfy all properties in Φ, which contradicts the fact that all these properties have probability 1 under σ. From the fact that no path under σ hits a state that cannot reach T_Φ, it follows by an easy induction that no finite trajectory under σ hits any bad state. That is, under σ all trajectories stay in the sub-MDP M''. Since every property $\psi_i \in \Psi$ has probability $\Pr_u^\sigma(\psi_i) > 0$ and almost all trajectories that satisfy ψ_i and Φ must hit a state of T_{R_i} (property (II) above), it follows that M'' contains some state of T_{R_i} for each $\psi_i \in \Psi$. Thus the algorithm returns Yes.

[5] Actually these sets are all computed together: we compute maximal *closed* components of the MDP, determine the properties that each component *favors* (see Def. 4.1 of [CY98]), and tag each state with the sets for which it is a target state.

[6] By "reach", we mean that starting at the state $v = v_0$, there a sequence of transitions $(v_i, \gamma, p_i, v_{i+1}) \in \delta$, $p_i > 0$, such that $v_n \in T_\Phi$ for some $n \geq 0$.

In the other direction, suppose that the algorithm returns Yes. First, note that for all states v of M'', and all enabled actions $\gamma \in \Gamma_v$ in M'', all transitions $(v, \gamma, p, v') \in \delta$, $p > 0$ of M' must still be in M'' (otherwise, γ would have been removed from Γ_v at some stage using rule 3(b)). On the other hand, some states may have some missing actions in M''. Next, note that all bottom strongly connected components (bscc's) of M'' (to be more precise, in the underlying one-step reachability graph of M'') contain a state of T_Φ (if $\Phi = \emptyset$ then all states are in T_Φ), for otherwise the states in these bsccs would have been eliminated at some stage using rule 3(a).

Define the following strategy σ which works in two phases. In the first phase, the trajectory stays within M''. At each control state take a random action that remains in M'' out of the state; the probabilities do not matter, we can use any non-zero probability for all the remaining actions. In addition, at each state, if the state is in T_Φ or it is in T_{R_i} for some property $\psi_i \in \Psi$, then with some nonzero probability the strategy decides to terminate phase 1 and move to phase 2 by switching to the strategy μ_Φ or μ_{R_i} respectively, which it applies from that point on. (Note: a state may belong to several T_{R_i}'s, in which case each one of them gets some non-zero probability - the precise value is unimportant.)

We claim that this strategy σ meets the desired requirements - it ensures probability 1 for all properties in Φ and positive probability for all properties in Ψ. For each $\psi_i \in \Psi$, the MDP M'' contains some state of T_{R_i}; with nonzero probability the process will follow a path to that state and then switch to the strategy μ_{R_i} from that point on, in which case it will satisfy ψ_i (property (I) above). Thus, all properties in Ψ are satisfied with positive probability.

As for Φ (if $\Phi \neq \emptyset$), note that with probability 1 the process will switch at some point to phase 2, because all bscc's of M'' have a state in T_Φ. When it switches to phase 2 it applies strategy μ_Φ or μ_{R_i} for some $R_i = \Phi \cup \{\psi_i\}$, hence in either case it will satisfy all properties of Φ with probability 1. □

5 Quantitative multi-objective model checking.

Theorem 4.

(1.) *Given an MDP M, an initial state u, and a quantitative multi-objective query B, we can decide whether there exists a strategy σ that satisfies B, and if so construct such a strategy, in time polynomial in $|M|$.*

(2.) *Moreover, given ω-regular properties $\Phi = \langle \varphi_1, \ldots, \varphi_k \rangle$, we can construct an ϵ-approximate Pareto curve $P_{M_u,\Phi}(\epsilon)$, for the set of achievable probability vectors $U_{M_u,\Phi}$ in time polynomial in M and in $1/\epsilon$.*

Proof. (Sketch.) For (1.), by the discussion in Section 2, we only need to consider extended achievability queries, $B \equiv \bigwedge_{i=1}^{k'} \Pr_u^\sigma(\varphi_i) \geq r_i \wedge \bigwedge_{j=k'+1}^{k} \Pr_u^\sigma(\varphi_j) > r_j$, where $k \geq k' \geq 0$, and for a vector $r \in (0,1]^k$. Let $\Phi = \langle \varphi_1, \ldots, \varphi_k \rangle$. We are going to reduce this multi-objective problem with objectives Φ to the quantitative multi-objective reachability problem studied in Section 3. From our reduction,

both (1.) and (2.) will follow, using Corollary 1. As in the proof of Theorem 3, we will build on constructions from [CY98]: form the MDP M' consisting of the product of M with the naive determinizations of the automata A_i for the properties $\varphi_i \in \Phi$. For each subset $R \subseteq \Phi$ we determine the corresponding subset T_R of target states in M'.[7]

Construct the following MDP M''. Add to M' a new absorbing state s_R for each subset R of Φ. For each state u of M' and each maximal subset R such that $u \in T_R$ add a new action γ_R to Γ_u, and an new transition $(u, \gamma_R, 1, s_R)$ to δ. With each property $\varphi_i \in \Phi$ we associate the subset of states $F_i = \{s_R \mid \varphi_i \in R\}$. Let $\overline{F} = \langle \Diamond F_1, \ldots, \Diamond F_k \rangle$. Let u^* be the initial state of the product MDP M'', given by the start state u of M and the start states of all the naively determinized A_i's. Recall that $U_{M_u,\Phi} \subseteq [0,1]^k$ denotes the achievable set for the properties Φ in M starting at u, and that $U_{M''_{u^*},\overline{F}}$ denotes the achievable set for $\overline{F}$ in M'' starting at u^*.

Lemma 3. *$U_{M_u,\Phi} = U_{M''_{u^*},\overline{F}}$. Moreover, from a strategy σ that achieves r in $U_{M_u,\Phi}$, we can recover a strategy σ' that achieves r in $U_{M''_{u^*},\overline{F}}$, and vice versa.*

It follows from the Lemma that: there exists a strategy σ in M such that $\bigwedge_{i=1}^{k'} \Pr_u^\sigma(\varphi_i) \geq r_i \wedge \bigwedge_{j=k'+1}^{k} \Pr_u^\sigma(\varphi_j) > r_j$ if and only if there exists a strategy σ' in M'' such that $\bigwedge_{i=1}^{k'} \Pr_{u^*}^\sigma(\Diamond F_i) \geq r_i \wedge \bigwedge_{j=k'+1}^{k} \Pr_{u^*}^\sigma(\Diamond F_j) > r_j$. Moreover, such strategies can be recovered from each other. Thus (1.) and (2.) follow, using Corollary 1. □

6 Concluding remarks

We mention that although our quantitative upper bounds use LP methods, in practice there is a way to combine efficient iterative numerical methods for MDPs, e.g., based on value iteration, with our results in order to approximate the Pareto curve for multi-objective model checking. This is because the results of [PY00] for multi-objective LPs only require a black-box routine that optimizes (exactly or approximately) positive linear combinations of the LP objectives. We omit the details of this approach.

An important extension of the applications of our results is to extend the asymmetric assume-guarantee compositional reasoning rule discussed in Section 2 to a general compositional framework for probabilistic systems. It is indeed possible to describe symmetric assume-guarantee rules that allow for general composition of MDPs. A full treatment of the general compositional framework requires a separate paper, and we plan to expand on this in follow-up work.

Acknowledgements. We thank the Newton Institute, where we initiated discussions on the topics of this paper during the Spring 2006 programme on Logic and Algorithms. Several authors acknowledge support from the following grants: EPSRC GR/S11107 and EP/D07956X, MRL 2005-04; NSF grants CCR-9988322, CCR-0124077, CCR-0311326, and ANI-0216467, BSF grant 9800096, Texas ATP grant 003604-0058-2003, Guggenheim Fellowship; NSF CCF-04-30946.

[7] Again, we don't need to compute these sets separately. See Footnote 5.

References

[Clí97] J. Clímaco, editor. *Multicriteria Analysis*. Springer-Verlag, 1997.

[CMH06] K. Chatterjee, R. Majumdar, and T. Henzinger. Markov decision processes with multiple objectives. In *Proc. of 23rd Symp. on Theoretical Aspects of Computer Science*, volume LNCS 3884, pages 325–336, 2006.

[CY95] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[CY98] C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. *IEEE Trans. on Automatic Control*, 43(10):1399–1418, 1998.

[Ehr05] M. Ehrgott. *Multicriteria optimization*. Springer-Verlag, 2005.

[EKVY07] K. Etessami, M. Kwiatkowska, M. Vardi, & M. Yannakakis. Multi-Objective Model Checking of Markov Decision Processes. Fuller version of this conference paper with proofs. `http://homepages.inf.ed.ac.uk/kousha/homepages/tacas07long.pdf`

[Fur80] N. Furukawa. Characterization of optimal policies in vector-valued Markovian decision processes. *Mathematics of Operations Research*, 5(2):271–279, 1980.

[Gho90] M. K. Ghosh. Markov decision processes with multiple costs. *Oper. Res. Lett.*, 9(4):257–260, 1990.

[GLS93] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 2nd edition, 1993.

[Hen83] M. I. Henig. Vector-valued dynamic programming. *SIAM J. Control Optim.*, 21(3):490–499, 1983.

[Put94] M. L. Puterman. *Markov Decision Processes*. Wiley, 1994.

[PY00] C. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Proc. of 41st IEEE Symp. on Foundations of Computer Science*, pages 86–92, 2000.

[Var85] M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. of 26th IEEE FOCS*, pages 327–338, 1985.

[Whi82] D. J. White. Multi-objective infinite-horizon discounted Markov decision processes. *J. Math. Anal. Appl.*, 89(2):639–647, 1982.

[WT98] K. Wakuta and K. Togawa. Solution procedures for multi-objective Markov decision processes. *Optimization. A Journal of Mathematical Programming and Operations Research*, 43(1):29–46, 1998.